

# Android-Workshop zur Vertiefung der Kenntnisse bezüglich Datenstrukturen und Programmierung in der Studieneingangsphase

**Manuel Froitzheim, Nadine Bergner, Ulrik Schroeder**

Lehr- und Forschungsgebiet Informatik 9

RWTH Aachen University

Ahornstr. 55

52074 Aachen

manuel.froitzheim@rwth-aachen.de

{bergner,schroeder}@informatik.rwth-aachen.de

**Abstract:** Die Studieneingangsphase stellt für Studierende eine Schlüsselphase des tertiären Ausbildungsabschnitts dar. Fachwissenschaftliches Wissen wird praxisfern vermittelt und die Studierenden können die Zusammenhänge zwischen den Themenfeldern der verschiedenen Vorlesungen nicht erkennen. Zur Verbesserung der Situation wurde ein Workshop entwickelt, der die Verbindung der Programmierung und der Datenstrukturen vertieft. Dabei wird das Spiel Go-Moku<sup>1</sup> als Android-App von den Studierenden selbständig entwickelt. Die Kombination aus Software (Java, Android-SDK) und Hardware (Tablet-Computer) für ein kleines realistisches Softwareprojekt stellt für die Studierenden eine neue Erfahrung dar.

---

<sup>1</sup> Spielregeln für ein Spielbrett mit acht mal acht Spielfeldzellen:

Die Spieler haben das Ziel als erster eine Reihe mit fünf Steinen der eigenen Farbe zu bilden. Die Steine müssen in einer Linie liegen, die sich horizontal, vertikal oder diagonal auf dem Spielbrett befindet. Zu Beginn des Spiels erhält jeder Spieler 32 Spielsteine in einer der beiden verfügbaren Farben. Ein Spieler setzt am Anfang einen Spielstein und anschließend wird abwechselnd jeweils ein Spielstein auf ein freies Feld des Spielfelds gelegt. Jeder Spieler versucht fünf Steine nebeneinander zu legen und den Gegner am Erreichen dieses Ziels zu hindern, indem der Bau einer Reihe durch blockierende Spielsteine verhindert wird. Längere Spielsteinreihen des Gegners stellen eine Gefahr dar, weil das Risiko auf eine fertige Reihe steigt und die Gefahr einer Niederlage wächst.

# 1 Motivation

An der RWTH Aachen University–und in ähnlicher Form auch an zahlreichen weiteren Hochschulen in Deutschland [HSS05] [HKLL10] [KKM13] [NAL+13] [IdSH13]–haben sich drei Problemfelder im Informatik Bachelor Studium herausgebildet:

1. Hohe Abbruchquote in den ersten Studiensemestern:  
Die Studieneingangsphase stellt die größte Hürde für die Studierenden der Informatik dar. [Wei10] Dies wird auch in der hohen Abbruchquote der ersten Semester deutlich. [He10] Eine geeignete Förderung kann die Studierenden in dieser Phase unterstützen und helfen die ersten Hürden des Informatikstudiums erfolgreich zu meistern.
2. Problematik der Verknüpfung verschiedener Themenfelder:  
Die Studierenden besuchen bereits in den ersten beiden Semestern verschiedene fachwissenschaftliche Vorlesungen aus den unterschiedlichen Bereichen der Informatik. In den Grundvorlesungen werden unabhängig voneinander Theorien und Konzepte aus den Disziplinen praktische, technische und theoretische Informatik vermittelt. Eine Verknüpfung des neu erworbenen Inselwissens der einzelnen Disziplinen oder auch praktische Anwendungsmöglichkeiten des erlernten Wissens sind in den Grundvorlesungen kaum in den regulären Veranstaltungen zu realisieren. Die Verknüpfung des Wissens aus den verschiedenen Vorlesungen und das Erlernen der Zusammenhänge sind für die Studierenden bei einer Fördermaßnahme anzustreben.
3. Verknüpfung Vorwissen von Programmierung und Datenstrukturen:  
Die Studierenden besuchen in der Studieneingangsphase, die die ersten beiden Semester umfasst, zum einen eine Programmierungsvorlesung zur Einführung in verschiedene Sprachen der objektorientierten, funktionalen und logischen Programmierung. Zum anderen eine Vorlesung „Datenstrukturen und Algorithmen“ zum Erlernen von Datenstrukturen (z. B. Listen und Bäume) und einfachen Algorithmen auf diesen (z. B. Sortieren und Suchen in einer Array Struktur). In den zugehörigen Übungen werden die Programmiersprachen Java, Haskell und Prolog zum Erlernen der Strukturen und für Beispiele verwendet. Allerdings sind die gewählten Ausschnitte sehr klein und eine Verzahnung der Grundstrukturen ist für Studierende kaum zu erkennen.

Die beschriebenen Problemfelder stellen die Ausgangslage für die Entwicklung des hier präsentierten Workshops dar. In diesem können Studierende des Fachbereichs Informatik an einem halben Tag unter Begleitung eine kleine alltagstaugliche App programmieren. [Hol12a] In diesem Anwendungs-Workshop werden Kenntnisse aus dem regulären Vorlesungsbetrieb aufgegriffen und in den kleinen Programmier-Projekten von den Studierenden vertieft und kombiniert. [Gla93]

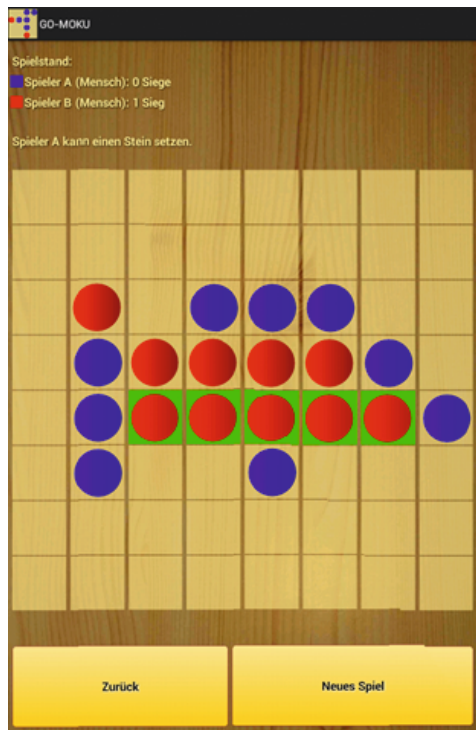


Abbildung 1: Screenshot der Android-App mit dem Go-Moku Spielbrett

Im Folgenden werden zunächst die Rahmenbedingungen und die Verbindung zu anderen Ausarbeitungen dieser Thematik dargestellt. Anschließend wird das an der RWTH Aachen University ausgearbeitete Workshop-Konzept beschrieben. Abschließend werden die Evaluierungsergebnisse des ersten Proberlaufs des Workshops dargestellt.

## **2 Hintergründe und Related Work**

### **2.1 Studiensituation (Rahmenbedingungen)**

In der Studieneingangsphase werden mehrere unabhängige Vorlesungen von den Studierenden besucht. Besonders hervorzuheben sind an der RWTH Aachen University die Vorlesungen „Einführung in die Programmierung“ und „Datenstrukturen und Algorithmen“. Viele Studierende haben bei den zugehörigen Klausuren Probleme diese beim ersten Versuch erfolgreich zu bestehen.

Es ist festzustellen, dass bei vielen Studierenden in der Studieneingangsphase ein Problem bei dem Umgang mit Datenstrukturen und einfachen Operationen auf diesen besteht. Besondere Aufmerksamkeit beim Workshop ist aus diesem Grund den mehrdimensionalen Arrays, Listen und Bäumen als Spezialform von Graphen, sowie den Operationen zur Bearbeitung dieser Strukturen zu schenken. Diese drei Grundstrukturen sollten unabhängig von der Programmiersprache von jedem Informatiker beherrscht werden, um effiziente und effektive Programme und informatische Lösungen zu gestalten.

Die Anwendung der Grundstrukturen in größeren Softwareprojekten wird in den Grundvorlesungen nicht vermittelt. Dies ist zum einen in der beschränkten Vorlesungszeit nicht möglich und zum anderen stehen in einer Vorlesung der Studieneingangsphase die Konzepte und nicht die konkrete Anwendung im Vordergrund. Trotzdem ist für die Studienmotivation der Studierenden unabdingbar die Zusammenhänge der Strukturen und möglichen praktischen Anwendungen frühzeitig zu erkennen.

### **2.2 Related Work**

An der RWTH Aachen University wurde im Rahmen des Projektes „IGaDtools4MINT“<sup>2</sup> ein vierstufiges Förderkonzept der Studierenden in der Studieneingangsphase entwickelt. [ABB+12] Die Förderung umfasst:

1. Außerschulische Förderung von Schülerinnen, Schülern und Studieninteressierten, zum Beispiel mithilfe des Schülerlabors Informatik InfoSphere<sup>3</sup>. [BHS12]

---

2 <http://www.igadtools4mint.de/> (zuletzt aufgerufen am 17.4.2014).

3 <http://schuelerlabor.informatik.rwth-aachen.de/> (zuletzt aufgerufen am 11.4.2014).

2. Anwendungsorientierter Vorkurs der Informatik als Studienvorbereitung der zukünftigen Studierenden. Kurz vor dem Studienbeginn können Studierende, deren Informatikkenntnisse Schwächen aufweisen, einen zweiwöchigen Vorkurs besuchen. Darin werden zum einen die Schwächen der Studierenden aufgearbeitet und zum anderen die Motivation für das Studium gesteigert. [Hol12b]
3. In der Studieneingangsphase kann mit ergänzenden Lehr-Lern-Angeboten gezielte Unterstützung den Studierenden geboten werden. Die Themen der Grundvorlesungen des Informatikstudiums können an sinnvollen Stellen vertieft, verknüpft und in einem realistischen Projekt erarbeitet werden. [ABB+12]
4. Integration von Gender- und Diversity-Aspekten in den regulären Lehrbetrieb.

Neben dem hier vorgestellten Workshop sind im Rahmen der dritten Stufe vier weitere Workshops zu unterschiedlichen Themenfeldern der Informatik entstanden. [BAT+14] Dieser Workshop reiht sich nach den Angeboten der drei Workshops im ersten Semester in die Mitte des zweiten Semesters ein.

### **3 Ausgestaltung des Workshops**

#### **3.1 Überblick über den Workshop**

Im zweiten Fachsemester ist durch das Modulhandbuch [RWT10] für den fachwissenschaftlichen Bachelor-Studiengang Informatik mit der Vorlesung „Datenstrukturen und Algorithmen“ ein Schwerpunkt auf selbige Themen gelegt. Der Schwerpunkt des Workshops liegt aus diesem Grund auf den Datenstrukturen. Des Weiteren werden grundlegende Programmierkenntnisse aufgegriffen, kleinere Algorithmen implementiert und ein Einblick in die Programmierung einer künstlichen Intelligenz gegeben.

Als Inhalt der App wurde für diesen Workshop das ursprünglich chinesische Brettspiel Go-Moku ausgewählt. Dieser Kontext wurde gewählt, da zum einen durch die Spielsituation die Motivation der Studierenden geweckt wird und zum anderen die gewünschten Datenstrukturen sinnvoll verwendet werden. Das Go-Moku Spiel eignet sich besonders gut, weil das Spiel auf einem Gitterraster beruht, welches als zweidimensionales Array repräsentiert werden kann. Im Rahmen des Workshops für das zweite Fachsemester werden Teile des Go-Moku Spiels für Android-Geräte programmiert. Der Nutzer der App soll sowohl gegen eine Person als auch gegen den Computer antreten können.

### 3.2 Inhaltliche und didaktische Entscheidungen

Das Ziel des Workshops besteht in der Vertiefung der Grundlagen der Datenstrukturen, ergänzt um die Vernetzung mit bestehenden Programmierkenntnissen und ersten Erfahrungen im Bereich der künstlichen Intelligenz. Im Workshop wird eine App für Tablet-Geräte mit einem Android-Betriebssystem erstellt, da so der Alltags- und Realitätsbezug für die Teilnehmer/innen direkt erfahrbar wird. Der zeitliche Umfang des Workshops ist auf circa vier Stunden beschränkt, damit dieser auch während der Vorlesungszeit an einem Nachmittag durchgeführt werden kann. In den verschiedenen Computerräumen der RWTH Aachen University, die für den Workshop genutzt werden, sind die Tische verankert in Reihen aufgebaut und handelsübliche Computer, auf denen die App programmiert wird, aufgestellt. Die Anordnung kann für den Workshop nicht verändert werden.

Die Lerngruppe setzt sich aus Studierenden mit unterschiedlichsten Kenntnisständen zusammen. Zum einen werden Studierende am Workshop teilnehmen, die mit der eigenständigen Programmierung von kleinen Programmen Probleme haben und zum anderen werden auch Studierende anwesend sein, die Interesse an der Android-Programmierung und wenig bis gar keine Probleme mit der Implementierung haben. Trotz dieser Heterogenität der Lerngruppe muss die Förderung der schwächeren Studierenden bei der Konzeption des Workshops im Fokus stehen. Eine gute Differenzierung und die angemessene Förderung eines jeden Workshopteilnehmers sind zu gewährleisten. [Fis73] Die Studierenden mit sehr guten Vorkenntnissen können zum Beispiel mittels Optimierungsaufgaben weitere und komplexere Herausforderungen erhalten. Wegen der Heterogenität sind regelmäßige Plenumsdiskussionen zu vermeiden, weil sonst die schnelleren Studierenden warten müssten und die langsameren gedrängt würden. Die Lerneinheit kann im ersten Teil Arbeitsphasen in der Gesamtgruppe beinhalten, diese sind im weiteren Verlauf des Workshops jedoch zu vermeiden. Aus diesem Grund gibt es beim hier dargestellten Workshop eine gemeinsame Einführung und Abschlussphase. Dazwischen können die Studierenden in einem angemessenen und selbstgewählten Tempo die unterschiedlichen Phasen durchlaufen, um im individuellen Lerntempo die Inhalte des Workshops zu erarbeiten. Damit die Studierenden während des Workshops über die Inhalte diskutieren können, ist für den gesamten Workshop Partnerarbeit, angelehnt an das Prinzip der Pair Programming, vorgesehen. Diese Sozialform bietet den Vorteil, dass die Studierenden sich ergänzen und gemeinsam über die Lösungen diskutieren können. Bei der Softwareentwicklung treten des Öfteren kleinere Fehler, meist

durch Tippfehler, auf. Die Korrektur kann viel Zeit in Anspruch nehmen. Wenn die Studierenden das Programm gemeinsam entwickeln, werden diese Fehler zügig gefunden und beseitigt. Des Weiteren werden auch Verständnisprobleme zügig bemerkt und entweder mit dem Partner oder mit einem Tutor geklärt.

### **3.3 Ablauf**

Für die Strukturierung des Workshops wurde auf das Vier-Phasen-Modell von Fässler zurückgegriffen [Fäs07]. Die Phasen werden mit „SEE“ (Entdeckung der Problemstellung), „TRY“ (erster Kontakt mit dem neuen Lerngegenstand), „DO“ (Entwicklung und Umsetzung eigener Ideen zur Strategie) und „EXPLAIN“ (Darstellung der gelernten Inhalte) bezeichnet. In jeder Phase wird das Wissensnetz zum Thema in unterschiedlichen kognitiven Dimensionen angesprochen. [Fäs07] Ein detaillierter Zeitplan kann der Abbildung 2 entnommen werden. Für den Workshop wurde den Studierenden eine Arbeitsmappe mit Arbeitsblättern, das Go-Moku Spiel in einer Brettspielvariante und eine Dokumentation des Grundgerüsts des Programmcodes zur Verfügung gestellt. Diese und weitere Materialien sind der Projektseite<sup>4</sup> zu entnehmen.

#### **SEE-Phase**

Nach einem Überblick über den Aufbau des Workshops steht in der SEE-Phase das Entdecken der Problemstellung im Vordergrund. Auf der kognitiven Dimension bedeutet dies die Betrachtung der Inhalte auf der deklarativen Ebene. [Fäs07]

Für den Workshop bedeutet dies, dass im ersten Abschnitt das Erlernen der Spielregeln des Go-Moku Spiels im Vordergrund steht. Für die weiteren Abschnitte des Workshops sind Kenntnisse der Spielregeln notwendig.

---

4 <http://learntech.rwth-aachen.de/Android DSAL> (zuletzt aufgerufen am 11.4.2014).


SEE	Begrüßung und Vorstellung Workshop 	25 Min.
	Spielregeln vorstellen Grundregeln des Spiels „Go-Moku“ vorstellen 	
	Go-Moku spielen Kennenlernen in der Gruppe und Festigung der Spielregeln 	
TRY	Vorstellen der Entwicklungsumgebung Erster Kontakt mit den Tablet-Computern, den Computern und der Software. 	60 Min.
	Programmgerüst Vorgegebenes Programmgerüst vorstellen 	
	Programmieren des Spielfelds Vorgegebene Struktur um die Klasse für das Spielfeld erweitern. 	
DO	PHASE 1: Positionierung per Zufall Die Steine werden per Zufall auf dem Spielbrett positioniert. 	125 Min.
	PHASE 2: Sieg des Gegners verhindern Die Züge des Gegners werden analysiert und der Sieg verhindert. 	
	PHASE 3: Siegen Das gesamte Spielbrett wird analysiert und ein optimaler Zug ausgewählt. 	
	Optimierung (Optional) Optimierung der Strategie nach eigenen Ideen. 	
EX- PLAIN	Wettbewerb Wettbewerb der verschiedenen Computerspieler. 	30 Min.
	Rückblick und Evaluation des Workshops Das Wissen festigen und Evaluation zur Optimierung für weitere Durchläufe. 	

Abbildung 2: Ablaufplan des Workshops

Die Studenten erhalten dazu die Spielregeln und eine physische Variante des Spielbretts mit Spielsteinen. In den jeweiligen Zweiergruppen können die Studierenden einige Male das Spiel durchführen. Währenddessen beantworten die Studierenden drei Leitfragen, damit eine Ergebnissicherung stattfindet und für die Entwicklung einer eigenen Strategie erste Ideen gesammelt werden.

### TRY-Phase

In der TRY-Phase, die einen Umfang von circa einer Stunde haben soll, erhalten die Studierenden die Möglichkeit, die ersten eigenen Erfahrungen mit der Programmierumgebung „Eclipse“ und den Aufgaben zu sammeln. Dazu sind stark angeleitete Aufgaben erstellt worden, damit die Studierenden zunächst



in kleinen Schritten an die unbekannte Programmierumgebung herangeführt werden. Wenn die Studierenden zu Beginn eine gewisse Grundsicherheit erhalten, sind komplexere Aufgaben im weiteren Verlauf möglich. In dieser Phase soll das deklarative Wissen aus der SEE-Phase mit dem prozeduralen Wissen verbunden werden. [Fäs07]

Konkret lernen die Studierenden die Arbeitsumgebung für den Workshop kennen. Die Entwicklungsumgebung „Eclipse“ wird in den ersten Semestern von den Studierenden wenig genutzt. Zur Android-Entwicklung sind diese Entwicklungsumgebung eine Standardanwendung und der kompetente Umgang mit den Grundfunktionen unerlässlich. In das vorgegebene Programmgerüst müssen sich die Studierenden erst einarbeiten, weil das gesamte Projekt umfangreicher ist, als ein durchschnittliches Projekt aus den ersten Studiensemestern. Das Programmgerüst besteht aus drei Quellcodepaketen. Die Aufgaben der Pakete teilen sich auf in die Strukturierung der graphischen Oberfläche, die Reaktionen der künstlichen Intelligenz und die Gestaltung des Spielablaufs inklusive der Gewinnerkennung.

Die Programmieraufgaben wurden für die Studierenden in kleinere Teilaufgaben gegliedert und im Sinne der bloomschen Taxonomie angeordnet. [BE72] Bei der ersten Programmieraufgabe wird die Struktur des Quellcodes im Aufgabentext klar vorgegeben und lediglich das Wissen um Standardkonstrukte wie Initialisierung eines Arrays wird benötigt.

In diesen ersten Schritten bei der Programmierung wird die Verwaltung für das Spielbrett erstellt. In einem zweidimensionalen Array müssen die Zustände der Spielbrettzellen gespeichert werden. Des Weiteren sind verschiedene Methoden für den Zugriff auf die Datenstruktur notwendig, damit ein sinnvoller Zugriff auf die Strukturen des Spielbretts gewährleistet werden kann. Das Verstehen, im Sinne der zweiten Klasse der bloomschen Taxonomie, einer sinnvollen Struktur für die Datenkapselung, muss von den Studierenden angewendet werden. Zum Beispiel die Entscheidung über den richtigen Sichtbarkeitsmodifikator (public, protected und private) obliegt den Studierenden und wird nicht vorgegeben. [Ull12]

Für die Implementierung der verschiedenen Strategien in der nächsten Phase wird eine Datenstruktur zur Speicherung der möglichen Positionen für einen Stein benötigt. Zum einen ist eine Arraystruktur ungeeignet, weil damit keine flexible Größenanpassung möglich ist. Zum anderen ist eine komplette Listenstruktur mit entsprechenden Zeigern für die Aufgabe überdimensioniert und in der beschränkten Zeit im Workshop kaum realisierbar. Der Kompromiss in dieser Situation ist eine erweiterte Arraystruktur mit Lese- und Schreibkopf zur Verwaltung der Positionen.<sup>5</sup> Die einzelnen Elemente des Ar-

rays werden von den Studierenden in einer Klasse BoardElement implementiert und mit C++. [Wol10] Getter- und Setter-Methoden kann der Zeilen- und Spaltenindex ausgewertet werden.

Das Ziel der TRY-Phase besteht darin, dass zum einen zwei Menschen auf dem Tablet-Gerät gegeneinander spielen können und die App einen möglichen Sieg erkennt. Zum anderen werden wichtige Grundlagen für die künstliche Intelligenz gelegt.

### **DO-Phase**

Die DO-Phase soll im Sinne des Konzepts von Fässler die längste Arbeitsphase in der Lerneinheit mit circa zwei Stunden bilden. In der DO-Phase sollen die Studierenden die Konzepte selbständig anwenden. Dazu sind Aufgaben mit wenig konkreten Vorgaben notwendig. In dieser Phase des Workshops eignen sich die Studierenden vor allem konditionales Wissen an. Konditionales Wissen ist die Ergänzung des deklarativen und des prozeduralen Wissens mit der Dimension zur richtigen Anwendung des Wissens. [Fäs07] Die Studierenden lernen selbständig den sinnvollen Einsatz von Methoden und Konzepten der Programmierung einzuschätzen. [HS08]

Aus diesen Gründen sind die Aufgaben in der DO-Phase deutlich freier formuliert als in den beiden vorherigen Phasen. Die Entwicklung der Spielstrategie erfolgt in drei Schritten und anschließend können weitere Optimierungen angebracht werden.

Die *erste Strategie* „Zufall“ dient zum einen zur Überprüfung des bisherigen Quellcodes und zum anderen erlangen die Studierenden ein Gefühl für das Vorgehen bei der Entwicklung einer Strategie. Hierbei soll per Zufall ein Stein auf eine freie Spielfeldzelle gesetzt werden. Die *zweite Strategie* „Sieg verhindern“ geht auf die aktuelle Spielsituation ein, indem längere Spielsteinketten des Gegners verhindert werden und wenn nötig Steine zur Blockade gelegt werden. Wenn keine Gefahr besteht, wird weiterhin eine Spielsteinzelle per Zufall ausgewählt und der Stein dorthin platziert. Die *dritte Strategie* „Siegen“ basiert auf einem vereinfachten Min-Max-Algorithmus. Der Algorithmus ermittelt eine optimale Spielstrategie für ein Nullsummenspiel mit perfekter Information. Für die nächsten Spielschritte wird ein Suchbaum mit den möglichen Spielsteinkonstellationen aufgebaut. Die eigenen Züge werden positiv und die des Gegners negativ bewertet. Die Werte der eigenen Züge werden maximiert und die Züge des Gegners minimiert. Die Bewertung der

---

5 Die Datenstruktur ist angelehnt an die Container-Klasse vector in der Programmiersprache.

Spielsituation erfolgt mit einer spielspezifischen Bewertungsfunktion. Zuerst wird analog zur vorherigen Strategie geprüft, ob längere Ketten des Gegners blockiert werden müssen. Wenn keine Gefahr droht, wählt der Algorithmus eine Position auf dem Spielbrett aus und legt einen Stein auf diese Position. Bei der Bewertung werden eigene längere Spielsteinketten ausgebaut und versucht einen Sieg zu erzielen. Bei der Optimierung können die Studierenden eigene Ideen einbauen und zum Beispiel den Min-Max-Algorithmus erweitern. Die Aufgaben der Studierenden in Bezug auf die einzelnen Strategien wurden nach der bloomschen Taxonomie gegliedert. [BE72] Die erarbeiteten Konstrukte aus der TRY-Phase müssen bei der ersten Strategie angewendet werden. Zum Beispiel ist die Array-Struktur notwendig um die freien Positionen auf dem Spielbrett zwischen zu speichern. Bei der Programmierung der zweiten Strategie sind die Analyse des Verhaltens des Programms und die Modifizierung des Quellcodes notwendig. Bei Schwierigkeiten muss das Studentenpaar in der Lage sein, das entsprechende Problem zu erklären und im Quellcode entsprechende Modifizierungen anzubringen. Bei der dritten Strategie müssen die Methoden der vorherigen Strategien genutzt werden. Die berechneten Werte aus den Methoden müssen für die Entscheidung zusammengesetzt werden. Der Min-Max-Algorithmus bietet eine Hilfestellung, um eine sinnvolle Struktur zu entwickeln. Bei der Optimierung der Strategie muss das Studentenpaar Entscheidungen auf Grundlage von eigenen Bewertungen treffen. Nachdem eine Idee für die Optimierung entstanden ist, sollte zunächst theoretisch überprüft werden, ob mit diesem Schritt eine Verbesserung erzielt werden kann.

Am Ende der DO-Phase sollte jedes Studentenpaar diese drei Strategien lauffähig für das Tablet-Gerät implementiert haben. Je mehr sinnvolle Optimierungsschritte erledigt werden, umso besser ist die Leistung des programmierten Computergegners.

### **EXPLAIN-Phase**

In der abschließenden EXPLAIN-Phase müssen die Lösungen von den Studierenden beschrieben werden. Das Erklären der eigenen Lösung erwirkt eine Metakognition des Wissens. Die Studierenden benötigen einen Überblick über die Inhalte des gesamten Workshops. Die Formulierung zur Erklärung der Inhalte verknüpft die verschiedenen erlernten Themen aus dem Workshop miteinander. [Fäs07]

Diese abschließende Phase wird für den Workshop mit einem kleinen Wettkampf der verschiedenen Implementierungen eingeleitet. Dieser Wettkampf fördert bei der Optimierung der eigenen Strategie die Motivation für

weitere Schritte. Die jeweils beste lauffähige Computerstrategie von zwei Studentengruppen kann auf ein Tablet-Gerät übertragen werden. Da jede Strategie unter Gesichtspunkten der Datenkapselung in einer eigenen Klasse implementiert wird, lassen sich die entsprechenden Klassen in einem Projekt zusammenführen. Anschließend können die beiden Strategien in der selben App ausgewählt werden und so automatisch gegeneinander spielen. Der Wettkampf fördert den Dialog zwischen den verschiedenen Gruppen und die Implementierung und Optimierungsschritte werden untereinander erklärt. Der Wettkampf ist angesichts der beschränkten Zeit eine gute Alternative zu einer Präsentation der verschiedenen Implementierungen. An dieser Stelle endet somit der vierstündige Workshop und die Teilnehmerinnen und Teilnehmer haben die Gelegenheit ihren Code zu speichern, so dass es ihnen möglich ist, bei Interesse zu Hause weiter daran zu arbeiten.

## **4 Evaluation**

### **4.1 Methodik**

Die Evaluierung des Workshops wurde mit einem Fragebogen vor und nach dem Workshop als quantitatives Pre-Post-Testdesign durchgeführt. Des Weiteren wurde gegen Ende des Workshops ein Teil der Studierenden informell nach ihren Eindrücken vom Workshop befragt. Diese qualitative Erhebung dient als Ergänzung des Fragebogens, weil bei der relativ kleinen Stichprobe von 20 Teilnehmer(inne)n keine signifikanten Zusammenhänge ermittelt werden können. Die Kombination aus den Methoden der quantitativen und qualitativen Erhebung lassen einige Rückschlüsse auf die Wirkung der Maßnahme zu. Darüber hinaus wurden die Arbeitsergebnisse der Studierenden ausgewertet.

### **4.2 Ergebnisse**

Am Workshop haben 20 Studierende teilgenommen. Beim ersten Erhebungszeitpunkt sind 20 Fragebögen ausgefüllt worden und zum zweiten Erhebungszeitpunkt sind 15 Fragebögen zurückgegeben worden<sup>6</sup>. Wegen der geringen Anzahl an Teilnehmer(inne)n ist bei der gesamten Auswertung zu beachten, dass die Ergebnisse nicht verallgemeinerbar sind. Für eine markante Veränderung des Gesamtergebnisses sind nur vereinzelte Ausreißer notwendig.

Die Mehrheit der Studierenden war im Alter von 19 bis 20 Jahren. Bei der Nationalität und der Muttersprache waren fast alle Teilnehmer gebürtige Deutsche mit einer Sozialisation im deutschen Sprachraum. Das Geschlechtsmerkmal war mit 95 % männlicher Teilnehmer und entsprechend nur 5 %, genau einer Studentin, weiblicher Teilnehmerinnen, nicht ausgewogen verteilt. Die angeworbenen Studierenden studierten fast alle im zweiten Fachsemester. Nur ein Student besuchte bereits das vierte Fachsemester.

Die Motivation während des Workshops war sehr hoch, obwohl es für den Workshop keinerlei Übungspunkte oder andere Vergütungen gab. Dies zeigt bereits wie gut das Thema des Workshops das Interesse der Studierenden wecken konnte. In den individuellen Befragungen wurde die Idee für ein Projekt, indem ein Spiel programmiert wird, unter Einsatz von Tablet-Geräten, positiv bewertet. Eine fertige Android-App mit einer graphischen Oberfläche ist für Studierende in den ersten Semestern interessanter als eine Konsolenausgabe auf dem Bildschirm, wie es in den meisten Vorlesungen üblich ist. Der Einsatz von mobilen Endgeräten und Computern zur Implementierung einer Android-App hat die Motivation gestärkt. Bei den Inhaltsfragen des Fragebogens konnte festgestellt werden, dass vor dem Workshop bei einigen Sprachkonstrukten Probleme bestanden. Diese Probleme konnten während des Workshops beseitigt werden, wobei entweder das Material selbst, der/die Partner/in oder ein/e Betreuer/in die nötige Hilfe gab, und traten im Fragebogen nach dem Workshop nicht auf. Die Anwendung der Konstrukte aus den Vorlesungen in einem Projekt und die praktische Anwendung, zum Beispiel von Getter- und Setter-Methoden, verdeutlichten die Relevanz der Inhalte. Die Studierenden gaben an, dass sie sich zum Teil die Anwendung dieser Methoden im Vorfeld als unnötig oder umständlich vorgestellt hatten.

## **5 Fazit und Ausblick**

Der erste Durchgang des Workshops wurde von den Studierenden positiv aufgenommen und evaluiert. Die 20 freiwilligen Teilnehmer/innen beim ersten Durchgang konnten für die Inhalte begeistert werden und waren in der Lage die Aufgaben in angemessener Zeit zu lösen. Den Evaluationsergebnissen zufolge konnte bei den 20 Teilnehmer(innen)n ein Erfolg verzeichnet werden.

---

6 Fünf Personen mussten aufgrund von Einschränkungen durch den öffentlichen Nahverkehr den Workshop frühzeitig verlassen.

In Zukunft ist es mit den erstellten Materialien ohne größeren Aufwand möglich, diesen Workshop zu wiederholen. Mit geringem personellem Aufwand kann ein längerfristiger Erfolg erzielt werden und den nächsten Studentengenerationen bei der Motivation für die Inhalte der Informatik geholfen werden. In Kombination mit weiteren ähnlichen Workshops zu anderen Themengebieten der Informatik kann darüberhinaus ein nachhaltiger Erfolg erzielt werden. Ein Einsatz der Materialien bei begabten Schüler(inne)n ist eine weitere Option. Wenn die Bearbeitungszeit für das Arbeitsmaterial des Workshops in angemessener Form erweitert wird, können auch Schüler/innen aus einem Leistungskurs oder einem Grundkurs mit guten bis sehr guten Noten diesen Workshop durchlaufen und gute Ergebnisse erzielen. Dies ist zum Beispiel als Modul im Schülerlabor Informatik InfoSphere der RWTH Aachen University möglich, also auf der Stufe 1 des Maßnahmenkatalogs (vgl. Abschnitt 2). Ein Einsatz der Materialien in einem Projektkurs in der gymnasialen Oberstufe ist ebenfalls eine Option. Wenn mehr Zeit zur Verfügung steht, können die Android-Grundlagen gemeinsam mit den Schülerinnen und Schülern erarbeitet werden und zum Beispiel beginnend mit der Entwicklung der graphischen Oberfläche kann in das Projekt eingestiegen werden. Insgesamt gibt es zahlreiche Möglichkeiten die für den Workshop entwickelten Materialien zusätzlich einzusetzen. Mit kleinen Modifizierungen bzw. didaktischen Reduktionen ist der Kreis der Interessenten deutlich erweiterbar. In der vorliegenden Form sind Studierende in der Studieneingangsphase und Oberstufenschüler/innen mit guten Leistungen in der Informatik eine geeignete Zielgruppe.

## Literaturverzeichnis

- [ABB+12] Rebecca Apel, Tobias Berg, Nadine Bergner, Mohamed Amine Chatti, Jan Holz, Carmen Leicht-Scholten und Ulrik Schroeder. Ein vierstufiges Förderkonzept für die Studieneingangsphase in der Informatik. In: Peter Forbrig, Detlef Rick und Axel (Hrsg.) Schmolitzky, Hrsg., *HDI 2012 – Informatik für eine nachhaltige Zukunft*, S. 143–148. Universitätsverlag Potsdam, Potsdam, 2012.
- [BAT+14] Tobias Berg, Rebecca Apel, Hendrik Thüs, Ulrik Schroeder und Carmen Leicht-Scholten. Vielfalt in der Informatik – Ergebnisse des Forschungsprojektes IGaDtools4MINT. In: Carmen Leicht-Scholten und Ulrik Schroeder, Hrsg., *Informatikkultur neu denken – Konzepte für Studium und Lehre. Integration von Gender und Diversity in MINT-Studiengängen*, S. 5–38. Springer-Verlag, Heidelberg, 2014.
- [BE72] Benjamin Samuel Bloom und Max D. Engelhart. *Taxonomie von Lernzielen im kognitiven Bereich*, Jg. 35. Beltz, Weinheim [u. a.], 1972.
- [BHS12] Nadine Bergner, Jan Holz und Ulrik Schroeder. Concept of an Extracurricular Learning Environment for Computer Science. In: M. Knobelsdorf und R. (Hrsg.) Romeike, Hrsg., *Proceedings of the 7th Workshop in Primary and Secondary Computing Education*, S. 26–33. NY: ACM, New York, 2012.
- [Fäs07] Lukas Fässler. *Das 4-Schritte-Modell: Grundlage für ein kompetenzorientiertes E-Learning von Lukas Emanuel Fässler*. Eidgenössische Technische Hochschule Zürich, Zürich, 2007.
- [Fis73] Margret Fischer. *Differenzierung im Schulunterricht*, Jgg. 43. Beltz, Weinheim und Basel, 1973.
- [Gla93] Dieter Glaap. Kreativ Arbeiten mit Computern. In: Tilman Ernst, Hrsg., *Computerspiele*, S. 137–145. Bundeszentrale für politische Bildung, Bonn, 1993.
- [He10] Ulrich Heublein et al. *Ursachen des Studienabbruchs in Bachelor- und in herkömmlichen Studiengängen: Ereignisse einer bundesweiten Befragung von Exmatrikulierten des Studienjahres 2007/08*. Forum Schule, Hannover, 2010.
- [HKLL10] Birgit Hilliger, Peter Kossack, Uta Lehmann und Joachim Ludwig. Die bedarfsorientierte Weiterentwicklung von Studieneingangsphasen: Ein Projektbericht aus der Universität Potsdam. In: Christa Cremer-Renz, Gustav-Wilhelm Bathke, Ludwig Huber, Clemens Klockner, Jürgen Lühje, Beate Meffert, Klaus Palandt, Ulrich Teichler, Wolff-Dietrich Webler und André Wolter, Hrsg., *Das Hochschulwesen*, S. 134–139. UniversitätsVerlagWebler, Bielefeld, 2010.

- [Hol12a] Jan Holz. IGaDtools4MINT: Integration von Gender and Diversity in MINTStudiengängen an Hochschulen, 2012.
- [Hol12b] Jan Holz. Progra Workshops: Eigene Apps im ersten Semester, 2012.
- [HS08] Anita Woolfolk Hoy und Ute Schönplüg. *Pädagogische Psychologie*. Pearson Studium, München und Boston [u. a.], 10. Auflage, 2008.
- [HSS05] Ulrich Heublein, Robert Schmelzer und Dieter Sommer. Studienabbruchstudie 2005: Die Studienabbrecherquoten in den Fächergruppen and Studienbereichen der Universitäten and Fachhochschulen. *Kurzinformation HIS*, (1): 1–38, 2005.
- [IdSH13] Susanne In der Smitten und Ulrich Heublein. Qualitätsmanagement zur Vorbeugung von Studienabbrüchen. *Zeitschrift für Hochschulentwicklung*, 8(2): 98–109, 2013.
- [KKM13] Martina Kaiser, Cornelia Kellermann und Susanne Matthes. Fordern und Fördern in der Studieneingangsphase. In: Georg Simon Ohm, Hrsg., *Tageungsband zum 1. HDMINT Symposium 2013*, S. 144–157. DiNa-Sonderausgabe, Nürnberg, 2013.
- [NAL+13] Manfred Nagl, Mostafa Akbari, Thiemo Leonhardt, Marlin Frickenschmidt und Svenja Schalthöfer. Mentoring in der Studieneingangsphase–Erfahrungen aus der Informatik der RWTH Aachen, 2013.
- [RWT10] RWTH Aachen University. Prüfungsordnung für den Bachelor-Studiengang Informatik der Rheinisch-Westfälischen Technischen Hochschule Aachen: BPO Informatik, 2010.
- [Ull12] Christian Ullenboom. *Java ist auch eine Insel*. Galileo Press, Bonn, 10. Auflage, 2012.
- [Wei10] Karsten Weihe. *Studieneingangsphase: Einsichten und Empfehlungen des Fakultätstags Informatik*. TU Darmstadt, Darmstadt, 2010.
- [Wol10] Jürgen Wolf. *C++ von A bis Z: Das umfassende Handbuch*. Galileo Press, Bonn, 2., aktual. Aufl., 1., korr. Nachdr., Auflage, 2010.